

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-196650

(43)Date of publication of application : 14. 07. 2000

(51)Int. Cl. H04L 12/46

H04L 12/28

(21)Application number : 11-363638

(71)Applicant : LUCENT TECHNOL INC

(22)Date of filing : 22. 12. 1999

(72)Inventor : KODIALAM MURALIDHARAN  
SAMPATH  
LAU WING CHEONG  
YAN ANLU

(30)Priority

Priority number : 98 218576

Priority date : 22. 12. 1998

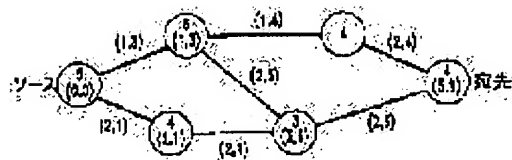
Priority country : US

### (54) CONDITIONAL SHORTEST ROUTING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To solve the shortest route within a network under an additional delaying condition by obtaining the shortest cumulative delay to a source from a destination, generating a cumulative delaying label to a destination node (k) from a node based on this and deciding an administrative weight as a link scale based on this delaying result.

SOLUTION: Dijkstra SPF is executed from a source again. The administrative weight(AW) is used as the link scale (metric) for deciding the shortest route. In this case, each node is labeled by accumulative AW from the source and a (Ds, i) label expressing accumulative delay from the source of the adjacent node (j) of a permanently labeled node (i) is updated only when  $(D_s, i + d_{j,i} + \delta i, i)$  satisfies the delay restriction of end to end such as 10. It is known that a route passing through a node 6 and a node 4 has delay being 11 and exceeds delay restriction. On the other hand, a route passing through the node 6 and a node 3 from the node 5 has total route delay being 9.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-196650

(P2000-196650A)

(43) 公開日 平成12年7月14日 (2000.7.14)

(51) Int.Cl.

識別記号

F I

キーワード (参考)

H 0 4 L 12/46  
12/28

H 0 4 L 11/00

3 1 0 C

審査請求 未請求 請求項の数13 O L (全 8 頁)

(21) 出願番号 特願平11-363638

(22) 出願日 平成11年12月22日 (1999.12.22)

(31) 優先権主張番号 09/218576

(32) 優先日 平成10年12月22日 (1998.12.22)

(33) 優先権主張国 米国 (US)

(71) 出願人 590092698

ルーセント テクノロジーズ インコーポ  
レーテッドアメリカ合衆国, 07974-0636 ニュージ  
ャーシー, マレイ ヒル, マウンテン ア  
ヴェニュー 600(72) 発明者 ムラリドハラン サンパス コディアラム  
アメリカ合衆国 07748 ニュージャーク  
イ, マールボロ, エア ドライブ 17

(74) 代理人 100064447

弁理士 関部 正夫 (外11名)

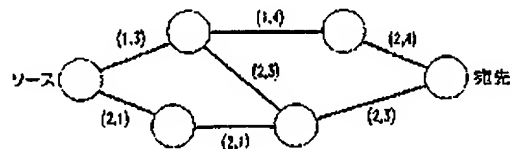
最終頁に続く

(54) 【発明の名称】 条件付き最短経路ルーティング法

(57) 【要約】

【課題】 本発明は、条件付き最短経路ルーティング法に関する。

【解決手段】 高速データ網、例えば、PNNIプロトコルを用いる網において、ノード間で許容される遅延制約の下で、ルーティングをリアルタイムにて遂行する本発明による方法は、2フェーズアルゴリズムを採用し、各フェーズにおいてDijkstraのアルゴリズムが用いられる。第一のフェーズにおいては、Dijkstra SPFアルゴリズムを用いて、宛先からソースへの最短累積遅延が求められ、これに基づいて、ノードjから宛先ノードkへの累積遅延ラベルが生成される。次に、この遅延結果が、第二のフェーズにおいて用いられる。より具体的には、第二のフェーズにおいては、Dijkstra SPFアルゴリズムを用いて、リンク尺度としての管理値m (AM) が第一のフェーズにおいて得られた結果に基づく修正の条件下で決定される。



(2)

特開2000-196650

1

2

## 【特許請求の範囲】

【請求項1】 網内のソースから宛先への経路を選択するための方法であって、この網が、複数のノードを含み、各ノードがリンクによって他の複数のノードに接続されており、各リンクが少なくとも第一と第二のリンクパラメータを反映する重みを持ち、前記ソースと宛先を接続する経路が複数のリンクから成り、これら経路が経路内のリンクに対する前記第二のパラメータの累積値に関する制約を持ち、この方法が：

a. 前記宛先から前記ソースへの複数の経路について、前記第二のパラメータに基づく最小の重みを持つ経路を識別する動作を前記網内の複数の経路に対して遂行し、これから、各ノードjから宛先kに対して前記第二のパラメータに対する重みを識別するラベルを生成するステップ、

b. 前記ソースから前記宛先への複数の経路について、前記ソースからの最小の重みを持つ経路を識別する動作を、前記第一のパラメータをリンク尺度として用いて遂行するステップ、および

c. 前記第二のパラメータに対する前記累積値に関する制約が違反されるようなノードを除去するステップ、から構成されることを特徴とする方法。

【請求項2】 前記網の動作がプライベート網インタフェース（PNNI）プロトコルに従うことを特徴とする請求項1の方法。

【請求項3】 前記第一のパラメータが管理重み（AW）であることを特徴とする請求項1の方法。

【請求項4】 前記第二のパラメータがリンク遅延であることを特徴とする請求項2の方法。

【請求項5】 前記宛先からの複数の経路について遂行される前記最小の重みを持つ経路を識別する動作が、Dijkstra SPFアルゴリズムを用いて達成されることを特徴とする請求項1の方法。

【請求項6】 前記ソースからの複数の経路について遂行される前記最小の重みを持つ経路を識別する動作が、Dijkstra SPFアルゴリズムを用いて達成されることを特徴とする請求項5の方法。

【請求項7】 さらに、前記網内の前記ソースノードから各フィジカルな宛先ノードへの複数の経路について、前記ステップa、b、およびcを反復するステップを含むことを特徴とする請求項6の方法。

【請求項8】 さらに、第一の宛先ノードの後の複数の宛先ノードについて、前記ソースから一つあるいは複数の他の宛先への経路に関する情報を用いて、前記第一の宛先の後の前記各ノードへの許容できるノードを決定するステップを含むことを特徴とする請求項7の方法。

【請求項9】 さらに、前記ソースからの複数の経路について遂行される前記最小の重みを持つ経路を識別する動作から、前記第一のパラメータに対する累積値が所定の最大値を超える経路を削除するステップを含むことを

特徴とする請求項1の方法。

【請求項10】 前記ソースからの複数の経路について最小の重みを持つ経路を識別する動作を遂行するステップが、（i）ノードiをソースノードからの累積管理重み（AW）にてラベリングするサブステップと、（i+1）前記ソースからの累積遅延（DS,i）を前記宛先からの複数の経路について最小の重みを持つ経路を識別する動作の際に得られた前記第二のパラメータに対する値に基づいて追跡するサブステップから成ることを特徴とする請求項1の方法。

【請求項11】 前記削除ステップが、パーマメントにラベルされたノードiの隣接ノードjを、 $(D_{i,j} + d_{i,j} + \delta_{i,j})$ が前記第二のパラメータに関する指定されるエンド・ツー・エンド（ノード間）遅延制約より小さな場合のみラベリングするサブステップを含み、ここで、 $\delta_{i,j}$ は、ノードiとjの間のリンクに対する前記第二のパラメータの値を表し、 $d_{i,j}$ は、ノードjとkとの間のリンクに対する前記第二のパラメータに対する値を表すことを特徴とする請求項10の方法。

【請求項12】 前記第二のパラメータがリンク遅延であることを特徴とする請求項10の方法。

【請求項13】 前記第二のパラメータがリンク遅延であることを特徴とする請求項11の方法。

## 【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、高速データ網において用いるためのデータ経路をルーティングする（決定する）ための方法およびシステムに関する。より詳細には、本発明は、例えば、PNNI（プライベート網インタフェース）プロトコルを用いる高速データ網において用いるための好ましいルーティング経路を選択するための方法およびシステムに関する。

【0002】

【従来の技術】高容量データ網、例えば、高速非同期転送モード（ATM）スイッチを用いるデータ網は、データの高速配信のみでなく、多様なサービス品質（QoS）を保証することを約束する。このような高速データ網においてノードを相互接続する経路を確立するに当たっての重要な要素は、高効率で信頼できるルーティング機構である。

【0003】近年、ATM Forum（ATMフォーラム）は、幾つかのPNNIプロトコルに関して標準化を行なった。これによって定義されるように、これらPNNIプロトコルは、ノード間で複数の異なるQoSパラメータを持つ接続を設定および解放することを可能にするシグナリングおよびルーティングプロトコルを提供する。これらPNNIプロトコルは、さらに、ノード間でトポロジ状態情報を交換することを可能にする。これらの詳細については、“Traffic Management Specification”, ATM Forum, 95-0013, R1.0, Feb. 1995; および Dykeman, D., and M. Coquen, “PNNI S

(3)

特開2000-196650

3

pecification", Ver. 1.0, ATM Forum, AF-PNNI-0055.000, May, 1996を参照されたい。

【0004】PNNIプロトコルの多くの実現が提唱されており、これらは、多様な性能およびQoS特性を待つ。多くの網、例えば、PNNIプロトコルを用いる網において考慮すべき重要な事項として、所望のノード間を、最小の網資源を用いて相互接続することがある。より具体的には、ノード間を、最短の相互接続経路を用いて相互接続することが要請される。加えて、選択されたノード間の伝送を低いあるいは最小の遅延時間にて達成することが、特に、(マルチメディア用途などの)多くの時間に依存する用途において以前にも増して重要となっている。短縮されたあるいは最小の遅延に基づくQoS基準は(所定のサービス品質を所定の遅延時間内で提供すること)は、従って、クリティカルなアプリケーション(遅延に敏感な用途)に対しては極めて重要であり、サービスプロバイダにとっても、これを達成することは、他社との競争上の優位を確保するための重要な要素となり得る。

【0005】

【発明が解決しようとする課題】選択されたノード間の最短経路(あるいは最小定義コストを持つ経路)を選択するための多くの解法(探索アルゴリズム)が存在し、これらの多くは、多数の相互接続されたノードを含む網においても用いることができる。これら多くの最短経路アルゴリズムの中でも、Dijkstraアルゴリズムは、特に卓越しており、良く知られている。Dijkstraその他のアルゴリズムに基づく解法は、しばしば、様々なQoS基準に關しての幾つかの条件が導入できるように修正される。Dijkstraアルゴリズムと対応するコードは、既に、豊富な経験を持ち、多くのケースにおいて信頼できることが実証されており、様々なリアルタイム網においても採用されている。従って、現存のコードを再利用することと、現存の最短経路解法と遅延条件付きの解法との間の機能的互換性を維持することは、非常に重要である。ただし、当分野においては周知のように、追加の遅延条件の下で網内の最短経路を解く問題は、いわゆる“NP-困難”問題であり、これは、リアルタイム網の背景では、現在の解法を用いて扱うことは困難であることが知られている。

【0006】

【課題を解決するための手段】従来の技術のこれら制約の克服および技術的な進歩が後に詳細に説明する本発明の幾つかの実施例に従う網ルーティング技法によって達成される。本発明の一つの実施例によると、網内のソースノードの所で、網内の他の各ノードへの最短重み経路が、許容される遅延制約を満たすという条件の下で、決定される。好ましくは、これら決定は、2つのフェーズから成る方法にて、各フェーズにおいて修正されたDijkstraのアルゴリズムを用いて遂行される。

4

【0007】第一のフェーズにおいては、Dijkstra SPFアルゴリズムを用いて、宛先からソースへの最短累積遅延が求められ、これに基づいて、ノードjから宛先ノードkへの累積遅延ラベルが生成される。次に、この遅延結果が、第二のフェーズにおいて用いられる。つまり、第二のフェーズにおいては、Dijkstra SPFアルゴリズムを用いて、リンク尺度としての管理重み(AW)が、本発明のもう一つの面による修正の下で(第一のフェーズにおいて得られた結果に基づく修正の下で)、決定される。より詳細には、一つの実施例においては、(1)ノードjがソースノードからの累積管理重み(AW)にてラベルされ、(2)(ソースからの)フォワードAWラベリングの際に累積遅延( $D_{j,i}$ )が追跡され、(3)パーマメントにラベルされたノードjの隣接ノードiは、( $D_{j,i} + d_{i,i,j} + \delta_{i,i,j}$ )が指定されるエンド・ツー・エンド遅延制約以下である場合にのみラベリングされる(ここで、 $\delta_{i,i,j}$ は、ノードiとjとの間のリンクの遅延を表す)。

【0008】

【発明の実施の形態】以下に本発明を幾つかの実施例を付図の図面との関連で詳細に説明する。図1は、例えば、ATM網内のノードに対応する一例としてのノードの網を示す。各ソース(発呼)ノードは、制御プロセッサを備え、これは、(周知のように)網ノード状態情報を受信し、低コスト、すなわち、ほぼ最適な最短経路に関する分析を、最大累積遅延制約の条件下で遂行する。コストは、一例として、一部は、管理重み(administrative weight, AW)の観点から測定される。加えて、本発明のもう一面によると、コストは、一部は、リンク遅延の観点から測定される。以下では、本発明を、最初に、単一のソースが単一の宛先への低コストのルーティングを探索するユニキャスト動作(unicast operation)の観点から説明し、次に、単一のソースから複数の宛先へのマルチキャスト動作(multicast operation)の観点から説明する。

【0009】ルーティングは、ある与えられた呼に対して、ソースの所であるいはソースに代わって関連する別個のもしくは共有のプロセッサによって、遂行される。PNNI(private network node interface: プライベート網インタフェース)プロトコルによると、ノード間で通信されるリンク状態パラメータは、リンク尺度(link metrics)と、リンク属性(link attributes)の2つのタイプから成る。リンク尺度(メトリック)は、リンクの状態パラメータであり、このパラメータは、ある与えられた経路に沿っての全てのリンクパラメータの結合(総和)であり、こうして、リンク尺度は、ある特定の経路がある与えられた接続に対して許容できるか否かを示すことができる。他方、リンク属性は、ある経路内のあるリンクの状態を反映するリンク状態パラメータである。本発明の一面においては、経路重みの決定において、リンク尺度として、管理重み(AW)の代わりに、経路遅延

(4)

特開2000-196650

5

(path delay) が用いられる。経路遅延は、ある経路内の全てのリンクの遅延の総和として取られる（測定される）。他の網ノードから受信された経路遅延情報が、好ましくは、ノードプロセッサの所に、テーブルの形式にて格納される。

#### 【0010】モデル

網ルーティング問題を、網モデルの観点からより厳密に特性化することは有効なことである。この目的のために、以下では、 $G=(V,E)$  が網の有向グラフ表現を表すものと想定し、このトリー内には、 $n$  個のノードと、 $m$  個のアーキ (arc) が存在するものと想定する。さらに、 $s \in V$  は、ソースを表し、 $t$  は、宛先を表すものと想定する。さらに、 $(i,j) \in E$  は、グラフ内の有向エッジ (アーキ) を表すものと想定する。さらに、 $w_{ij} \geq 0$  は、アーキ  $(i,j)$  の重みを表し、 $d_{ij} > 0$  は、アーキ  $(i,j)$  に沿って発生する遅延を表すものと想定する。さらに、 $P(i,k)$  は、ノード  $i$  からノード  $k$  へのセットの単純経路を表すものと想定する。さらに、経路  $P \in P(i,k)$  の重みは、 $w(P)$  によって表し、これは、その経路内のアーキの重みの総和として定義されるものと想定する。さらに、経路  $P \in P(i,k)$  の遅延は、 $d(P)$  によって表し、これは、経路内のアーキの遅延の総和として定義されるものと想定する。すると、ルーティング問題の目的は、ソース  $s$  から宛先  $t$  への最小の重みの経路を、その経路の遅延がある所定の値  $D$  を超えないという条件下で、探索することとなり、これは、以下のように表される：

【外1】

$$\min_{P \in P(i,t)} w(P)$$

制約下で、

$$d(P) \leq D.$$

\*

PHASE 1( $s, t, d$ )

#### • INITIALIZATION

$$U = V, \quad W = \emptyset, \quad \delta_s = 0, \quad \delta_j = \infty \quad \forall j \neq s$$

#### • ITERATIVE STEP

$$k = \text{Arg min}_{j \in U} \delta_j$$

$$U \leftarrow U \setminus \{k\} \text{ and } W \leftarrow W \cup \{k\}.$$

$$\text{For each } j \in U, (i, j, k) \in E, \text{ if } \delta_k + d_{kj} < \delta_j \text{ then } \delta_j = \delta_k + d_{kj}$$

#### • TERMINATION TEST

$$\text{Stop if } s \in W.$$

【0013】 $\delta_j$  は、ノード  $j$  から宛先  $t$  への最短経路長を表すことに注意する。 $\delta_s > D$  である場合は、この問題に対するフィジブルな解は、存在せず、このアルゴリズムは、終了する。他方、 $\delta_s < D$  である場合は、Dijkstra

6

\* 上述のように、この問題は、NP-完全問題 (NPクラスにある最も困難な問題) である。このため、以下では、経路の重みを、遅延制約が違反されることを回避しながら、最小化するためのヒューリスティック (発見的) アルゴリズムが展開され；このアルゴリズムが、(遅延制約に関して) フィジブルな (条件を満たす) 経路に、それが存在する場合は、必ず、辿り着くことが示され；さらに、本発明のアルゴリズムは、当然の結果として、低い重みの経路を与えることが示される。

#### 【0011】アルゴリズムの説明

このセクションでは、各ソースにおいて実行されるアルゴリズムについて説明する。最初に、ある特定の宛先への経路を求めるアルゴリズムについて説明し、次に、このアルゴリズムを、あるソースから全ての宛先への最短重み経路を計算するためのサブルーチンとして用いることについて説明する。

#### 【0012】単一シンク (Single Sink)

$s$  は、ソースノードを表すものとし、このセクションでは、ソース  $s$  からある与えられた宛先ノード  $t$  への最短重み経路を決定する方法について述べる。ノードプロセッサでのこのアルゴリズムの処理は、好ましくは、2つのフェーズにて遂行される。このアルゴリズムの第一のフェーズにおいては、宛先からソースへの複数の経路に対して、周知のDijkstraアルゴリズムが、アーキの長さとしてのに (として表現される) 遅延ラベルを用いて実行され、このアルゴリズムの第二のフェーズにおいては、フェーズ1の実行の結果として得られた (遅延の重を表す) 距離ラベルが用いられる。以下に、このアルゴリズムのフェーズ1を示す：

30 【外2】

kstraのアルゴリズムの修正バージョンであるフェーズ2のアルゴリズムが、ソース  $s$  から実行される。

【0014】アルゴリズムの説明を簡単にするために、エッジ  $(i,j)$  の一般化された重み  $w_{ij}$  を、以下のよう

(5)

特開2000-196650

7

8

に定義する:

【数1】

$$v_g \triangleq (w_g, d_g) \quad (1)$$

単純経路  $PEP(1, k)$  の一般化された重みは、 $v(P) \cdot (w(P), d(P))$  として定義される。

【0015】フェーズ2において、各ノードの先行ノードが、 $\pi[1]$ 、 $\forall i \in V$  内に維持され、2チューブル\*

**Definition 1 Define**

$$(a_i, \psi_i) < (a_j, \psi_j)$$

if

- $a_i < a_j$ , or
- $a_i = a_j$  and  $\psi_i < \psi_j$

Phase 2 of the algorithm is:

**PHASE 2 ( $s, t, w, d$ )**• **INITIALIZATION**1:  $U = V, W = \emptyset, a_s = \psi_s = 0, a_j = \psi_j = \infty \quad \forall j \neq s,$  $\pi[j] = NULL \quad \forall j \in V$ • **ITERATIVE STEP**2:  $k = \text{Arg min}_{i \in U} (a_i, \psi_i)$ 3:  $U = U \setminus \{k\}$  and  $W = W \cup \{k\}$ .4: 4.1: For each  $f \in U, (k, f) \in E$ 4.2: if  $(a_k + w_{kf}, \psi_k + d_{kf}) < (a_f, \psi_f)$ 4.3: if  $(\psi_k + d_{kt} + \delta_f \leq D)$ 4.4:  $a_f = a_k + w_{kf}$ 4.5:  $\psi_f = \psi_k + d_{kf}$ 4.6:  $\pi[f] = k$ • **TERMINATION TEST**5: Stop if  $t \in W$ .

【0017】ライン4. 3を省いた場合、上述のアルゴリズムは、単に、一般化された重みに関するDijkstraのアルゴリズムとなることに注意する。換言すれば、こ

\*ラベル  $(\omega, \psi)$  が、各ノードに対して維持される。ここで、 $\omega$  は、ソース  $s$  からあるフィジブルな経路に沿っての累積重みを表し、 $\psi$  は、ソース  $s$  からこれと同一の経路に沿っての遅延を表す。これら2チューブルに対して、辞書式順序を、以下のように定義する。

【0016】

【外3】

れは、遅延ラベル  $\psi$  をタイブレーカ（同等の場合の判断基準）として用いたときの重みに関するDijkstraのアルゴリズムである。Dijkstraのアルゴリズムを実現するた

めに用いられる全ての引数は、式(1)に定義される一般化された重み $w$ を用いた場合にも適用できるために、以下の補題(補助定理)が成り立つ。

【0018】補題1 Dijkstraのアルゴリズムを、 $G$ について、一般化された重み $w$ に関して実行した場合、終端において、以下となる：

【数2】

$$(w_i, \psi_i) = \min_{P \in P(s, i)} v(P)$$

【0019】以下の定理は、本発明のアルゴリズムの特性を要約する。

定理2 フェーズ2は、遅延制約を満足するような経路にて終端する(経路を与える)。加えて、遅延制約を満足するような、重みに関して(重みの点で)最短な複数の経路が存在する場合、このアルゴリズムは、その一つを見つける。

【0020】証明：最初に、 $\delta_s \leq D$ である場合は、フェーズ2が、遅延制約を満足する経路を見つけることを示す。ライン4. 3と4. 4から、ノード $i$ の重みラベル $w_i$ は、ノード $i$ がフィジブルな経路の上に存在する場合にのみ有限な値を取る(つまり、遅延制約を満たす $s$ から $i$ への少なくとも一つの経路が存在すること)がわかる。加えて、ライン3は、各反復において、集合 $U$ から一つの要素(元)を除去する。従って、このアルゴリズムがフィジブルな経路を見つけることに失敗すると仮定した場合は、このアルゴリズムの実行の際のどこかの時点で、 $w_i = \infty, \forall i \in U$ となるはずである(を持つはずである)。ただし、これは、以下の理由からあり得ない。つまり、集合 $W$ 内にあるノードが含まれる限り、そのノードを通るソースから宛先への遅延制約を満たす経路が存在する。 $s \in W, t \in U$ であるため(そうでなければ、このアルゴリズムは成功しているはずである)。 $\exists k \in W, i \in U, s, t$ エッジ $(k, i)$ は、フィジブルな経路上に存在する。 $k$ が $W$ 内に含まれていた場合は、 $w_k$ の値は、有限値に更新されているはずであり、これは、矛盾である。

【0021】この定理の第二の主張を証明するために、最初に、Dijkstraのアルゴリズムを、一般化された重み $w$ に関して実行することを考える。このアルゴリズムが、 $P(s, t) = (s, i_1, \dots, i_n = t)$ によって表されるソース $s$ から宛先 $t$ への単純経路にて終端する(を与える)場合は、この経路上の各ノードは、ラベル $(w_{i_k}, \psi_{i_k}), k = 1, \dots, n$ を持つ。補題1から、 $P(s, t)$ は、重みに関して(の点で)ソース $s$ から宛先 $t$ への複数の経路の中の最短経路であり、このような最短経路が複数個存在する場合は、 $P(s, t)$ は、これらの中で最小の遅延を持つこととなる。従って、遅延制約を満たし、かつ、重みに関して最短な、複数の経路が存在する場合は、 $P(s, t)$ は、これらの中の一つである。

【0022】次に、重みに関して最短な他のどの経路

も、 $P(s, t)$ と同一の遅延を持たないものと想定し、数学的帰納法を用いて、フェーズ2が終端した時点で、

a)  $P(s, t)$ 上のノード $i_1$ に対するラベルは、まだ $(w_{i_1}, \psi_{i_1})$ であることと；b) ソース $s$ から宛先 $t$ への経路は、 $P(s, t)$ として見つけられることを証明する。

【0023】 $P(s, t)$ は、フィジブルな経路であるために、ライン4. 3に関する条件は、アーク $(s, i_1)$ が探索されるとき、満足される。補題1から、アーク $(s, i_1)$ は、ソース $s$ から $i_1$ への複数の経路の中で最小の一般化された重み $w_{s, i_1}$ を持ち、このため、ノード $i_1$ は、パーマネントに $(w_{i_1}, \psi_{i_1}) = (w_{s, i_1}, d_{s, i_1})$ とラベルされ、この先行ノードはソース $s$ となる。

【0024】 $(w_{i_1}, \psi_{i_1})$ と同一の一般化された重みを持つソース $s$ から $i_1$ への経路が他にも存在することも考えられることに注意する。この場合は、ノード $i_1$ は、ソース $s$ からではなく、これら複数の経路の一つの最後のノード、例えば $P(s, i_1)$ の最後のノードからパーマネントにラベルされることとなる。ただし、この場合も、ノード $i_1$ に対するラベルの値は同一となり、ソース $s$ は、このときも、ノード $i_1$ の、恐らくは幾つかの中間の先行ノードを持つ、先行トリーの上に存在するために、経路 $P(s, i_1)$ を見つけることは、 $(s, i_1)$ を見つけることと、重みと遅延に関する限りは、なんの差も生じない。従って、この証明の残りの部分では、このような状況は、証明の一般性を失うことなく、無視される。

【0025】ノード $i_1$ は、ソース $s$ にパーマネントに接続されるため、このノードは、例えば、アルゴリズムの最初の反復において、ソース $s$ が集合 $U$ から除去されたとき、パーマネントなラベルを取得する。これらラベルの値は、 $k=1, \dots, k_0-1$ に対して、増加せず、 $(w_{i_k}, \psi_{i_k}) \leq (w_{i_{k+1}}, \psi_{i_{k+1}})$ であるために、 $(w_{i_1}, \psi_{i_1}) < (w_{i_2}, \psi_{i_2})$  (あるいは等価的に $(w_{i_1, 12}, d_{i_1, 12}) > (0, 0)$ ) のときは、ノード $i_1$ は、 $i_2, \dots, i_{k_0}$ の前に集合 $U$ から除去される。一方、 $(w_{i_1}, \psi_{i_1}) = (w_{i_2}, \psi_{i_2})$  (つまり、 $(w_{i_1, 12}, d_{i_1, 12}) = (0, 0)$ ) のときは、ノード $i_1$ がノード $i_2$ の前に集合 $U$ から除去される可能性があるが、ただし、これは、 $P(s, i_2) = (s, i_1, i_2)$ であるために、単に、同一の重みと遅延を持つソース $s$ からノード $i_2$ へのもう一つの経路が見つかったことを意味する。ここでも、このようなケースは、ノード $i_1$ がノード $i_2$ の前に集合 $U$ から除去される場合と、これら経路は重みと遅延に関する限り等価であるために、区別しない。このため、証明の以下の部分においては、一般性を失うことなく、 $(w_{i_1}, \psi_{i_1}) < (w_{i_2}, \psi_{i_2})$ であるものと想定する。

【0026】数学的帰納法の目的で、ノード $i_1, \dots, i_{k_0}$ が集合 $W$ に移され、 $(w_{i_1}, \psi_{i_1}), \dots, (w_{i_{k_0}}, \psi_{i_{k_0}})$ としてパーマネントにラベルされたものと想定する。さらに、ソース $s$ から $i_1$ への経路が $(s, i_1, \dots, i_{k_0})$ であ

(7)

特開2000-196650

11

ることが見つかり、 $i_k$ がノード $i_1, \dots, i_{k-1}$ の前に集合 $U$ から除去されたものと想定する。以下では、この想定の下で、ノード $i_{k+1}$ は、その経路上の次のノードであり、 $(w_{i_{k+1}}, \psi_{i_{k+1}})$ とラベルされることを示す。

【0027】ノード $i_k$ が集合 $U$ から除去されると、アーク $(i_k, i_{k+1})$ が探索される。 $s$ から $i_k$ への経路は、 $P(s, i)$ の一部であるために、 $i_{k+1}$ は、 $P(s, i)$ 上に存在し、ライン4. 3の条件が満足される。このため、 $i_{k+1}$ は、 $(w_{i_k}, \psi_{i_k}) + (w_{i_{k+1}}, d_{i_{k+1}}) = (w_{i_{k+1}}, \psi_{i_{k+1}})$ とラベルされ、先行ノードは $i_k$ となる。 $(w_{i_{k+1}}, \psi_{i_{k+1}})$ は、補題1から最小のラベルであるために、ノード $i_{k+1}$ は、ラベルを変更されることはない。加えて、 $i_{k+1}$ は、ノード $i_{k+2}, \dots, i_n$ の前に集合 $U$ から除去される。これは、 $i_{k+1}$ は、ノード $i_k$ が集合 $U$ から除去されるとき、パーマネントラベルを取得し、このラベルは、 $i_{k+1}, \dots, i_n$ のそれより小さなためである。こうして、この数学的帰納法の証明は完了する。

【0028】最後に、 $P(s, t)$ と同一の遅延を持つ。 \*  
MULTIPLE DESTINATIONS(s)

• INITIALIZATION

-  $G=V, F=\emptyset, M=\infty$

• ITERATIVE STEP

- Pick  $i \in G$

• PHASE 1 (s, t, d)

• PHASE 2 (s, t, w)

• TERMINATION TEST

- Stop if  $G \in \emptyset$  else go to the ITERATIVE STEP.

【0031】図1の一例としての網に戻り、以下では、これに上述のフェーズ1の処理を適用することを考える。図2との関連で説明すると、フェーズ1は、ノード $j$ から宛先 $k$ への最小の遅延ラベル $(d_{j,k})$ を持つ経路を、逆Dijkstra SPFアルゴリズムを用いて決定することから成る。ここでは、ソースから宛先までの遅延は、10ユニット以下に制限されるものと想定される。リンクパラメータは、(リンク $AW$ , リンク遅延)として表現される。例えば、図2の最も上の経路では、リンク $AW$ , 遅延)チュープルは、(1, 4)となる。

【0032】図3は、フェーズ1の処理の結果を、フェーズ2の処理に適用する様子を示す。より詳細には、Dijkstra SPFが、再び、今回は、ソースから実行される。管理重み $(AW)$ が、“最短(shortest)”経路の決定するためのリンク尺度(メトリック)として用いられるが、ただし、ここでは、(i)各ノードは、ソースから

12

\* 重みの点で最短な他の経路が存在し、このような代替経路が、 $P(s, t)$ の代わりに、見つけれられる可能性があるが、これら経路は、重みと遅延に関する限り等価である。

【0029】マルチシンク(Multiple Sink)の場合

この場合は、PNNIにおける目的は、あるソース $s$ から全ての他のノードへの最短経路を決定することとなる。これは、フェーズ1とフェーズ2を、ソースから各シンクに対して実行することで行なわれる。ただし、このアルゴリズムがソースから宛先 $t$ への最短重み経路を見つけるために実行された場合、これがソースから他の宛先への最短経路に関する情報を与えることがある。より具体的には、集合 $F$ 内の全ての宛先に対して、条件付き最短経路問題に対する最適解が得られることがある。マルチシンクの場合は、この事実が利用される。以下にマルチシンクに対するアルゴリズムを示す：

【0030】

【外4】

の累積 $AW$ にてラベルされ、(i)パーマネントにラベルされたノード $i$ の隣接ノード $j$ のソースからの累積遅延を表す $(D_{i,j})$ ラベルは、 $(D_{i,i} + d_{i,j} + \delta_{i,j})$ がエンド・トゥ・エンド(ノード間)遅延制約(この例では10)を満足する場合のみに更新される。図3からは、図3の網表現の上側の外周に沿っての経路は、11なる遅延を持ち、遅延制約を超えることがわかる。他方、ソース5からのノード6と3を通る経路は、9なる総経路遅延を持つ。ここで、ノードの円内のチュープルは、それぞれ、リンク $AW$ の総和と、リンク遅延の総和を表す。

【0033】以上の説明から、当業者においては、現在利用可能なDijkstraアルゴリズム技法を、既存のプログラムコードも含めて、本発明の教示に従って、データ網のノードにおいて、PNNIプロトコルの通常の使用によって得られるリンクデータを用いてリアルタイムルーティングを遂行するように、適合できることを理解できると



(8)

特開2000-196650

13

14

考える。

【0034】ここに開示されるアルゴリズムおよび方法を採用する本発明の方法およびシステムの使用することで、ノードプロセッサあるいは代替プロセッサは、エンド・トゥ・エンド（ノード端間）遅延制約を満足する、網を通じての一つあるいは複数のフィジブルな経路を、少なくとも一つのこのような経路が存在する場合は、見つける。

【0035】AM最短経路（管理査みの点で最短な経路）が遅延制約を自動的に満たす場合は、その経路が本発明の教示を用いて選択される。上で説明の方法では、一つのソースから全ての宛先への最短経路解を求めるためには、Dijkstraアルゴリズムが $n+1$ 回用いられたが、当業者においては理解できるように、状況によっては、宛先までの遅延ラベル $d_{i,j}$ をノード間で交換すること \*

\*で、計算負荷を低減することもできる。さらに、上の詳細な説明は、FNNIプロトコルの背景で行なわれたが、本発明は、様々な網プロトコルを用いる様々なタイプの網にも有効に適用できるものである。

【0036】当業者においては、本発明の上述の様々な教示から、他の修正および拡張が考えられると思われるが、これらも全て、本発明の精神および範囲に含まれるものである。

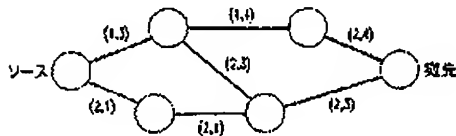
【図面の簡単な説明】

【図1】本発明の一つの実施例を解説するための一例としての網を示す図である。

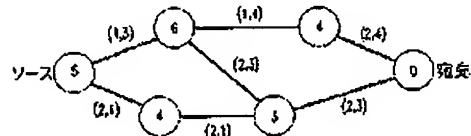
【図2】図1の網に適用したときの本発明の一つの実施例のフェーズ1を理解するのに役立つ図である。

【図3】図1の網に適用したときの本発明の一つの実施例のフェーズ2を理解するのに役立つ図である。

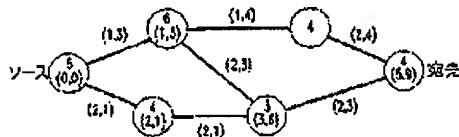
【図1】



【図2】



【図3】



フロントページの続き

(72)発明者 ウィン チェン ロー  
アメリカ合衆国 07724 ニュージャージー  
ィ、イースタウン、ビクトリア ドライ  
ヴ 40

(72)発明者 アンル ヤン  
アメリカ合衆国 07724 ニュージャージー  
ィ、イースタウン、ウェッジウッド サ  
ークル 66